**第一题源码：**

#include<iostream>

using namespace std;

int CommonYear[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 }; //平年每月天数,闰年CommonYear[1] = 29

//[函数名] isLeapYear

//[功能] 判断是否是闰年

//[参数] int year：年份

//[返回值] true:是闰年false:否

bool isLeapYear(int year) {

bool OK = 1;

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) OK = 1;

else OK = 0;

return OK;

}

class Date {

public:

Date(int y = 0, int m = 1, int d = 1) :year(y), month(m), day(d) {

if (isLeapYear(y)) CommonYear[1] = 29; //判断当前年是否是闰年

else CommonYear[1] = 28;

if (m < 1 || m > 12 || d < 1 || d > CommonYear[m - 1]) {

cout << "日期非法！" << endl;

abort();

}

}

Date(Date& dat) {

this->year = dat.year;

this->month = dat.month;

this->day = dat.day;

}

void showDate() {

cout << "日期为:" << year << "年" << month << "月" << day << "日" << endl;

}

Date operator + (int days);

Date operator - (int days);

int operator - (Date& b);

private:

int year,month,day;

};

int Date::operator - (Date& b) { //支持自动比较两个日期大小返回差值

int count = 0;

Date high; //high为两日期中较大的

Date temp; //temp为两日期中较小的

bool OK = 1; //1为b在\*this的日期之前。如this（2001，8，30），b（2000，1，1）。0相反

int y, m, d;

//判断二者先后

y = this->year - b.year; //若b在this之前，此值为正

m = this->month - b.month;

d = this->day - b.day;

if (y < 0) OK = 0;

if (y == 0 && m < 0) OK = 0;

if (y == 0 && m == 0 && d < 0)OK = 0;

if (y == 0 && m == 0 && d == 0) {

return 0;

}

if (OK == 1) { temp = b; high = \*this; }

else { temp = \*this; high = b; }

//进行计数；

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断当前年是否是闰年

else CommonYear[1] = 28;

while (temp.year != high.year || temp.month != high.month || temp.day != high.day) {

if (temp.day < CommonYear[temp.month - 1]) { //天数小于当前月份天数，可以自加

temp.day++; //日期自加1

count++; //相差天数自加1

}

else if (temp.day == CommonYear[temp.month - 1] && temp.month != 12) { //非12月的月底

temp.month++;

temp.day = 1;

count++;

}

else if (temp.day == CommonYear[temp.month - 1] && temp.month == 12) { //12月月底

temp.year++;

temp.month = 1;

temp.day = 1;

count++;

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断今年（下一年）是否是闰年

else CommonYear[1] = 28;

}

}

return count;

}

Date Date::operator + (int days) {

Date temp(\*this);

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断当前年是否是闰年

else CommonYear[1] = 28;

while (days > 0) {

if (temp.day < CommonYear[temp.month - 1]) { //天数小于当前月份天数，可以自加

temp.day++; //日期自加1

days--; //需要天数自减1

}

else if (temp.day == CommonYear[temp.month - 1] && temp.month != 12) { //非12月的月底

temp.month++;

temp.day = 1;

days--;

}

else if (temp.day == CommonYear[temp.month - 1] && temp.month == 12) { //12月月底

temp.year++;

temp.month = 1;

temp.day = 1;

days--;

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断今年（下一年）是否是闰年

else CommonYear[1] = 28;

}

}

return temp;

}

Date Date::operator - (int days) {

Date temp(\*this);

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断当前年是否是闰年

else CommonYear[1] = 28;

while (days > 0) {

if (temp.day > 1) { //天数大于1天，可以自减

temp.day--; //日期自减1

days--; //需要天数自减1

}

else if (temp.day == 1 && temp.month != 1) { //非一月初

temp.month--; //月份自减1

temp.day = CommonYear[temp.month - 1]; //日期变为当前（前一个月）月的总天数

days--;

}

else if (temp.day == 1 && temp.month == 1) { //一月初

temp.year--;

temp.month = 12;

temp.day = 31;

days--;

if (isLeapYear(temp.year)) CommonYear[1] = 29; //判断今年（下一年）是否是闰年

else CommonYear[1] = 28;

}

}

return temp;

}

int main() {

int y1, m1, d1, y2, m2, d2;

cout << "请输入date1的参数(年 月 日):";

cin >> y1 >> m1 >> d1;

cout << "请输入date2的参数(年 月 日):";

cin >> y2 >> m2 >> d2;

Date dat1(y1, m1, d1),dat2(y2,m2,d2);

Date dat3;

int nSelect,days;

cout << "请选择功能:" << endl << "1.返回Date1加天数days后得到的日期" << endl <<

"2.返回Date1减去天数days后得到的日期" << endl << "3.返回两日期相差的天数" << endl << "其他数字键退出" << endl;

cin >> nSelect;

switch (nSelect) {

case 1: {

cout << "请输入天数days:";

cin >> days;

dat3 = dat1 + days;

dat3.showDate();

break;

}

case 2: {

cout << "请输入天数days:";

cin >> days;

dat3 = dat1 - days;

dat3.showDate();

break;

}

case 3: {

int count;

count = dat1 - dat2;

cout << "相差" << count << "天";

break;

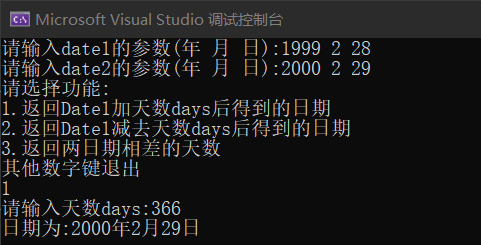
}

}

return 0;

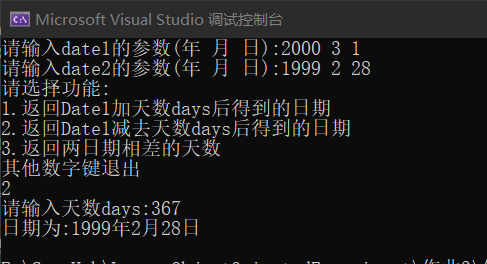
}

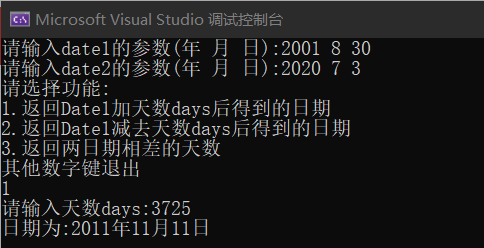
**第一题测试结果：**

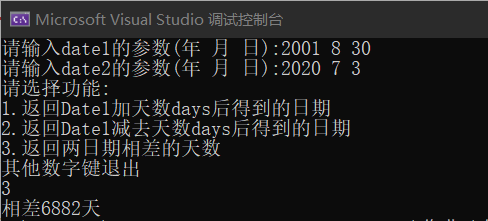
****

**首先测试了增加天数，加了366天，由1999年2月28日可知，确实应该是2000年2月29日。之后测试了用2000年3月1日减去367天，由之前增加天数推算，应该正好是99年2月28日，测试次信息则是为了测试闰年。**

**之后我第一次测试就直接用我的生日作为起点，随便输了个3725去增加天数，结果得到了个2011年11月11日（雾…），虽然和网上找的计算器算出来一模一样，一遍就过了，但是我真的好伤心，这冥冥之中是不是预示着什么（K%Z8KRF)XN}]~EGEQY{J2AR）。之后算了一下我生日到今天写完一共多少天，答案6882，和网上的答案运算算出来一样。应该算是过了吧（哭兮兮）。**



****

****

****

**第二题源代码：**

#include<iostream>

#define PI 3.1415926

using namespace std;

class body {

public:

body(double dat):data(dat){}

virtual double getSize() = 0;

virtual double getVolumn() = 0;

protected:

double data; //作为球的半径、正方体的边长或圆柱体低面圆的半径

};

class sphere :public body {

public:

sphere(double r) :body(r) {};

double getSize() {

return 4 \* PI \* data \* data;

}

double getVolumn() {

return 3.0 / 4 \* PI \* data \* data \* data;

}

};

class cube :public body {

public:

cube(double length) :body(length) {};

double getSize() {

return 6 \* data \* data;

}

double getVolumn() {

return data \* data \* data;

}

};

class cylinder :public body {

public:

cylinder(double r,double h) :body(r),height(h) {};

double getSize() {

return 2 \* PI \* data \* data + 2 \* PI \* data \* height;

}

double getVolumn() {

return data \* data \* PI \* height;

}

private:

double height;

};

int main() {

sphere s1(7.5);

cube c1(6.3);

cylinder cy1(5.5, 10);

cout << "球的表面积为：" << s1.getSize() << endl;

cout << "球的体积为：" << s1.getVolumn() << endl;

cout << "正方体的表面积为：" << c1.getSize() << endl;

cout << "正方体的体积为：" << c1.getVolumn() << endl;

cout << "圆柱体的表面积为：" << cy1.getSize() << endl;

cout << "圆柱体的体积为：" << cy1.getVolumn() << endl;

return 0;

}

**第二题测试结果：**



**第三题源码：**

#include<iostream>

using namespace std;

template<class T>

void show(T a[], int size) {

cout << "前" << size << "项已经排序完成，结果为：" << endl;

for (int i = 0; i < size; i++) {

cout << a[i] << " ";

}

cout << endl;

}

template<class T>

void sort(T a[], int size) {

T temp;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (a[j] > a[j + 1]) {

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

show(a, size);

}

int main() {

int arr1[] = { 7,2,4,5,7,9,10,14,6,3,1 };

double arr2[] = { 1.7,6.5,3.4,2.5,6.3,2.6,7.3,5.4,3.2,2.7,6.0 };

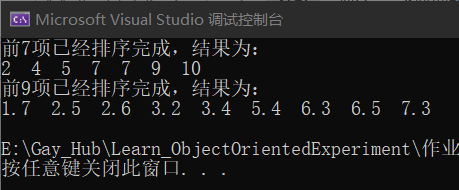
sort(arr1, 7);

sort(arr2, 9);

return 0;

}

**第三题测试结果：**



**第四题源码：**

#include<iostream>

using namespace std;

template<class T1,class T2>

class Test

{

public:

void SetData1(T1 val);

void SetData2(T2 val);

T1 GetData1();

T2 GetData2();

private:

T1 data1;

T2 data2;

};

template<class T1,class T2>

void Test<T1,T2>::SetData1(T1 val) { data1 = val; }

template<class T1,class T2>

void Test<T1, T2>::SetData2(T2 val) { data2 = val; }

template<class T1, class T2>

T1 Test<T1, T2>::GetData1() { return data1; }

template<class T1, class T2>

T2 Test<T1, T2>::GetData2() { return data2; }

int main() {

Test<int,double> t;

t.SetData1(13);

t.SetData2(17.5);

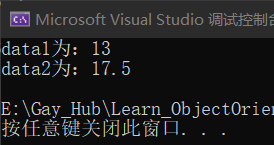
cout << "data1为：" << t.GetData1() << endl;

cout << "data2为：" << t.GetData2() << endl;

return 0;

}

**第四题测试结果：**

****

**第五题源码：**

#include <iostream>

using namespace std;

class Complex

{

public:

Complex(double re = 0, double im = 0):real(re), imag(im) {};

//重载<<运算符实现输出复数

friend ostream& operator <<(ostream& output, Complex& c);

//重载>>运算符实现输入复数

friend istream& operator >>(istream& input, Complex& c);

private:

double real;//实部

double imag;//虚部

};

istream& operator >>(istream& input, Complex& c)

{

int a, b;

char sign, i;

do{

cout << "请输入一个复数,以a+bi的形式输入:";

input >> a >> sign >> b >> i;

} while (!((sign == '+'|| sign == '-') && 'i' == i));

c.real = a;

c.imag = ('+' == sign) ? b : -b;

return input;

}

ostream& operator <<(ostream& output, Complex& c)

{

double num = c.imag;

if (num > 0){

output << c.real << "+" << c.imag << "i" << endl;

}

else{

output << c.real << c.imag << "i" << endl;

}

return output;

}

int main(){

Complex c1,c2;

cin >> c1;

cin >> c2;

cout << c1;

cout << c2;

return 0;

**}**

**第五题测试结果：**

